Review of Gualtiero Piccinini, Physical Computation: A Mechanistic Account

Gualtiero Piccinini, *Physical Computation: A Mechanistic Account*.
Oxford: Oxford University Press, 2015. ix + 313 pp.

The question of which physical systems compute is of pressing foundational import for the cognitive sciences in light of repeated claims that intelligence, cognition, or even consciousness reduce to, or supervene on, computational properties. Traditional philosophical approaches to this question have begun from the mathematical theory of computation, seeking a formal relationship between its abstract models, such as Turing machines, and physical systems. In contrast, Gualtiero Piccinini's *Physical Computation* develops a sustained answer grounded in the practices of computer science, in particular the details of computer hardware and architecture. The thoroughness with which Piccinini presents his view and the insight it brings to longstanding debates establish it as a gold standard against which future theories of concrete computation will be assayed.

Physical Computation comprises sixteen chapters, which divide thematically into three main topics: Chapters 1 through 4 establish the success conditions for an adequate theory of physical computation and demonstrate that previous accounts in the literature fail to satisfy them. Chapters 5 through 7 articulate Piccinini's positive theory, developing a teleological account of mechanistic explanation that culminates in a statement of the characteristic functional features of computational mechanisms. The remaining chapters, almost half the total length, apply this theory to a variety of examples and debates. While fourteen of the chapters draw on separately published work, there is much added value in the presentation of this material as an integrated whole, as it reveals the nuance and richness of a comprehensive view not fully accessible through its piecemeal precursors.

Piccinini's welcome decision to begin with an explicit discussion of the success conditions for his project serves both as orientation to his prior commitments and as touchstone for subsequent discussion. From the start, the guiding role of computer science is on display, for instance in the requirements that an adequate theory of physical computation be *objective*, *explanatory*, and able to successfully *differentiate* those systems that compute from those that do not. For Piccinini, the very existence of a science of computing rules out anti-realist views on which whether or not a system computes is a mere matter of perspective—desktops compute and rocks do not, and any account that cannot make this distinction (as early isomorphism-based accounts could not) is inconsistent with scientific practice (11, 34). Likewise, Piccinini takes semantic accounts, on which computation presupposes representation, to run afoul of the computer scientist's ability to build a computer that manipulates symbols arbitrarily and without semantic interpretation (35). He further requires that an adequate account explain (the possibility of) *miscomputation*, as well as correctly recover the technical *taxonomy* of types of computation (11–15).

Piccinini's positive account of mechanistic computation is embedded within a theory of teleological mechanisms developed in conjunction with Carl Craver and Corey Maley (Chapters 5 and 6). Chapter 5 rejects the autonomy of top-down functional analysis, arguing it should be construed as a means to "mechanism sketches" that must cohere with bottom-up investigation (75); consequently, functional analysis that does not identify its subcomponents with "structural" (read: *physical*, 80) entities is not properly explanatory. Chapter 6 develops a teleological alternative to etiological accounts of mechanism function. Identifying properties with causal powers (105), Piccinini singles out two "special" properties distinctive of organisms: homeostatic preservation, or *survival*, and *inclusive fitness*, noting they

both require the expenditure of energy. He then defines an *objective goal* as "that state toward which such a special property manifestation is directed, and which requires work on the part of the organism via particular mechanisms" (106); *subjective goals* are those "due to sentience or sapience" (116). A *teleological function* is then a "stable contribution" by some trait or artifact to an objective or subjective goal (108, 116).

With these preliminaries in place, a *physical computing system* may be defined as a mechanism with the teleological function of computing (121). Nevertheless, the success of this account does not turn primarily on the details of the teleological theory that precedes it; rather, the real work is in the notion of computation on offer:

> **Generic Computation**: the processing of vehicles by a functional mechanism according to rules that are sensitive solely to differences between different portions (i.e., spatiotemporal parts) of the vehicles.
>
> **Rule**: a mapping from inputs **I** (and possibly internal states **S**) to outputs **O**. (121)

This is essentially an elaboration of a syntax-based account of computation (44–7) that (a) abandons the constraint that syntactic relations exhibit the recursive structure found in language (46), and (b) replaces it with the requirement that the relevant relations be *medium independent*, that is, "can be defined independently of the media that implement them" (122). The demand for medium independence does crucial work, establishing multiple realizability (122–3), while ruling out putative counterexamples to non-semantic accounts, namely non-computational input-output systems (e.g. digestive tracts (146) or mouse traps (153)).

The remainder of the book demonstrates the power of this theory for resolving or clarifying muddled disputes about putative computational systems in nature. I'll briefly illustrate this power with two examples before returning to the notion of medium independence and its implications for a semantics-free account of computation.

Chapter 13 nicely dissolves many of the debates surrounding the relationship between neural networks and classical computation. For instance, a point of contention in this literature has been the claim that neural networks, and thus plausibly the brain, perform "massively parallel" computations—if they do, does this substantively challenge classical conceptions of computational explanation in psychology? Piccinini helpfully distinguishes the claim that "more than one computational operation" occurs within a specified time frame from the claim that "more than one instruction" is executed (215–6). The careful elucidation of computer architecture in the previous chapters reveals how standard digital computers are parallel in the first sense: typically, many Boolean circuits are operational at the same time, and in fact the syncing of these operations through a centralized clock is an essential feature of modern computer architecture (159–60, 169–72). Conversely, it is not clear that neural networks themselves perform parallel computations in the second sense. Piccinini argues that neural networks do not in general "execute" programs, since they do not structurally instantiate the sequence of steps that define a program (210–1); insofar as a (typical) neural network *instantiates* a computational process, it does so serially, as it directly transforms a single input into a single output (211, 216). Consequently, the issue of parallelism is a red herring; rather, the key question at stake is whether

computation in the brain exhibits non-classical features such as *trainability* or *continuous dynamics* (219–21, 223–4).

Piccinini's final topic is the "Physical Church-Turing Thesis," the claim that all physical computational systems are Turing-equivalent, i.e. compute at best the same class of functions computable by Turing machines. Traditionally, the PCT has been important as a question about the limitations of minds and the devices we build to aid them. More recently, some physicists have proposed that fundamental physical processes are computational, and thus the universe itself performs computations. However, success at computationally modeling a system does not necessarily imply it is best *explained* as performing computations (23, 69–71). If we are to assess whether a computational explanation is appropriate, we need a physical analog to the mathematical notion of an "effective procedure" that motivated Church and Turing themselves. Piccinini proposes a

> **Usability Constraint:** If a physical process is a computation, it can be used by a finite observer to obtain the desired values of a function. (250)

This constraint allows us to helpfully generalize (again) from the details of actual computers to other systems, now asking not how they are structured, but rather how one may interact with them. Claims that the universe as a whole, randomized processes (260–1), or systems involving arbitrarily precise values (259, 271) compute are quickly ruled out by this constraint—no finite user can exploit them for performing computations because she cannot reliably *initialize* them, *repeat* her calculations, or effectively *measure* their input and output values (251–5). The usability constraint also rules out specific suggestions for physical super-Turing computers, such as Hogarth's relativistic computer (266–70) or Siegelmann's analog networks (271).

I want to conclude by raising a worry for Piccinini's project and suggest a possible avenue for response. My concern is that the thoroughly non-semantic view Piccinini defends does not have the resources to fully resolve the questions about human cognition that motivated his investigation in the first place. We can see the shadow of this worry already in the *usability* response to the PCT. One reason we care about whether there are super-Turing computations in nature is because we want to know whether the brain, and thus human cognition itself, might be super-Turing. This is not a question that can be resolved by appealing to a prior notion of a finite user, since it concerns the very nature of paradigmatic users themselves. In this case, I think the solution is straightforward: Piccinini's decomposition of the usability constraint into sub-demands for initializability, repeatability, and measurability of input–output values can, I think, be applied to cognitive processes without fear of circularity. Nevertheless, there are other questions for which the fix is not so easy.

Consider, for instance, the *Sieve Problem* that Haugeland (1980) raises against Fodor's flirtation with the idea that psychological explanation should be purely syntactic. Haugeland points out that one cannot reduce computation to rule governed interactions determined solely by "local syntactic properties," since then even the sorting of grains by a sieve counts as computational. In responding to similar examples, Piccinini appeals to the requirement that computation must respond differentially to medium independent features of inputs, i.e. differences between "spatiotemporal parts" only along some "specific dimensions of variance" (122, 176–7). For instance, the stomach, while instantiating an input–output function from foodstuffs to their chemical constituents, does so in a way inherently tied to

the chemical composition of the media it processes (146–7). But a sieve responds only to some dimensions of spatiotemporal variance (lengths or volumes) and not others (the chemical composition of sorted granules), its abstract rule to sort by magnitude may be instantiated in different devices (e.g. a centrifuge), and different media (prisms sort light by magnitude of wavelength). About the sieve problem in particular (176), Piccinini's implicit argument is that a sieve does not exhibit the right kind of complexity to be understood as a digital computer (177*f*). But mere sorting of magnitudes is not intuitively even a *generic* computational process, and if he embraced it as such, Piccinini would seem to violate his own requirement that his theory successfully differentiates those systems that compute from those that do not.

One response to these worries is to accept a tempered role for semantics in defining computation. Ironically, despite Piccinini's protest that semantics is irrelevant for individuating computations (31–44), his positive theory is suffused with semantic language. For instance, he requires that computations be defined over "vehicles" (121), a term motivated by semantic considerations, namely the need to distinguish *content* from its bearer, or *vehicle*. He is forced to use this term, rather than appealing to bare physical parts of a computational mechanism, precisely because computational properties are determined at an abstract, medium-independent level. Yet this abstract characterization of computational vehicles and the rules that manipulate them itself constitutes a kind of semantics, a point Piccinini himself develops at length as "internal semantics" (135–6, 173–5) and explicitly requires as part of his usability constraint (*definability*, 252). The most his earlier arguments against semantic approaches actually show is that the assignment of physical, external referents is irrelevant to computation, not that computation may be characterized without appeal to *any* semantic concepts. This conclusion should be unsurprising since the mathematical theory of computation constitutively appeals to representations, namely of abstract objects (numbers and functions over them), and computer scientists themselves traffic primarily in semantic objects, namely programming languages. My suggestion is that Piccinini embrace the software side of computer science as enthusiastically as he has embraced the hardware, allowing a more modest role for semantics in characterizing computation, one not prey to the objections he legitimately levels against earlier approaches, but that has the resources to avoid sieve problems and speak more directly to questions about the computational nature of cognition itself.

While the question at stake here is major, the room for maneuvering against Piccinini is slight, so comprehensive is his articulation and defense of the mechanistic account. This rigor of detail and breadth of application establish *Physical Computation* as essential reading for all those working on philosophy of computation, and recommend it strongly to philosophers of mind, physics, biology, or any other area in which appeals to concrete computation arise.

References

Haugeland, J. (1980) "Formality and Naturalism," *Behavioral and Brain Sciences* 3: 81–2.

Alistair M. C. Isaac
University of Edinburgh